

EL6 24352343

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Fast Dynamic Measurement of Connection Bandwidth

Inventor:

David del Val
Bill Schieffelbein
Troy Batterberry
Tom Brown

ATTORNEY'S DOCKET NO. MS1-542US

09636004-030500

1 **TECHNICAL FIELD**

2 This invention relates to dynamic detection of maximum bandwidth for a
3 connection between to entities on a network.
4

5 **BACKGROUND**

6 As the Internet has matured, the format characteristics of the content
7 available on the Internet have changed. Sound and video content is now mixed in
8 with the traditional textual content. However, this new content on the Internet
9 requires a greater connection speed (i.e., bandwidth) than was commonly available
10 a few years ago.

11 Fig. 1 illustrates an example of a typical Internet configuration. It includes
12 a server (such as media server 20), which is coupled to the Internet 30. The server
13 typically includes one or more physical server computers 22 with one or more
14 physical storage devices and/or databases 24. On the other side of an Internet
15 transmission is a client 90, which is connected via one of many available Internet
16 Service Providers (ISPs) 80. Herein, a server is a network entity that sends data
17 and a client is a network entity that receives data.

18 Cloud 30 is labeled the Internet, but it is understood that this cloud
19 represents that portion of the Internet that does not include the server, client's ISP,
20 and the client. Inside such cloud are the routers, transmission lines, connections,
21 and other devices that more-often-than-not successfully transmit data between
22 clients and servers. Inside exemplary Internet cloud 30 are routers 32-44; two
23 satellite dishes 46 and 50; and a satellite 48. These represent the possible paths
24 that a data packet may take on its way between the server and the client.
25

005009-400950

1 **Bandwidth**

2 Bandwidth is the amount of data that can be transmitted in a fixed amount
3 of time. For example, bandwidth between media server 20 in Fig. 1 to media
4 client 90 is calculated by the amount of data (e.g., 1000 bits) that may be
5 transmitted between them in a unit of time (e.g., one second).

6 As shown in Fig. 1, a transmission over the Internet travels across multiple
7 links before it reaches its destination. Each link has its own bandwidth. Like a
8 chain being only as strong as its weakest link, the maximum bandwidth between
9 server 20 and client 90 is the link therebetween with the slowest bandwidth.
10 Typically, that is the link (such as link 82 in Fig. 1) between the client 90 and its
11 ISPs 80. That slowest bandwidth is the maximum de facto bandwidth.

12 Herein, unless otherwise apparent from the context, references to
13 bandwidth between network entities (such as server 20 and client 90) is assumed
14 to be the maximum de facto bandwidth therebetween.

15 Bandwidth may also be called "connection speed", "speed", or "rate". In
16 references to bandwidth measured by bits per second, it may also be called "bit
17 rate" or "bitrate."

18
19 **Streaming Media**

20 Streaming is a technique for transferring multimedia data such that it can be
21 processed as a steady and continuous stream. Streaming technologies are
22 becoming increasingly important with the growth of the Internet because most
23 users do not have fast enough access to download large multimedia files quickly.
24
25

1 With streaming, the client browser or plug-in can start displaying the data before
2 the entire file has been transmitted.

3 For streaming to work, the client side receiving the data must be able to
4 collect the data and send it as a steady stream to the application that is processing
5 the data and converting it to sound or pictures. This means that if the streaming
6 client receives the data more quickly than required, it needs to save the excess data
7 in a buffer. If the data doesn't come quickly enough, however, the presentation of
8 the data will not be smooth.

9 Within the context of an audio and/or visual presentation, "media" and
10 "multimedia" are used interchangeably herein. Media refers to the presentation of
11 text, graphics, video, animation, and/or sound in an integrated way.

12 "Streaming media" is an audio and/or visual presentation that is transmitted
13 over a network (such as the Internet) to an end-user. Such transmission is
14 performed so that the presentation is relatively smooth and not jerky. Long pauses
15 while additional frames are being downloaded to the user are annoying to the user.
16 These annoyances encourage a user to avoid viewing future streaming media.

17 18 **Smoothly Transmitting Streaming Media**

19 Since the bandwidth determines the rate at which the client will receive
20 data, a streaming media presentation may only be presented at a rate no greater
21 than what the bandwidth allows. For example, assume media server 20 needs to
22 send data at 50Kbps to the client 90 in order to smoothly "play" a streaming media
23 presentation. However, the bandwidth between the client and server is only
24 30Kbps. The result is a jerky and jumpy media presentation.

1 In an effort to alleviate this problem, streaming media presentations are
2 often encoded into multiple formats with differing degrees of qualities. The
3 formats with the lowest quality (e.g., small size, low resolution, small color
4 palette) have the least amount of data to push to the client over a given time.
5 Therefore, a client over a slow link can smoothly present the streaming media
6 presentation, but the quality of the presentation suffers. The formats with the
7 highest quality (e.g., full screen size, high resolution, large color palette) have the
8 greatest amount of data to push to the client over a given time. Therefore, the
9 client with a fast link can smoothly present the streaming media presentation and
10 still provide a high quality presentation.

11 12 **Select-a-Bandwidth Approach**

13 When a server sends streaming media to a client, it needs to know what
14 format to use. Thus, in order to select the proper format, the server must to know
15 the bandwidth between the server and the client.

16 This easiest way to accomplish this is to ask the user of the client what their
17 bandwidth is. Since a client's link to the Internet is typically the bandwidth
18 bottleneck, knowing the bandwidth of this link typically indicates the actual
19 bandwidth.

20 Fig. 2 shows a cut-away 100 of a Web page displayed on a client's
21 computer. Inside the cut-away 100, is a typical user-interface 110 that may be
22 used to ask a user what their connection speed is. The user clicks on one of the
23 three buttons 112, 114, and 116 provided by the user-interface 110. If the user
24 clicks on button 112, the server delivers data from a file containing streaming
25

media in a format designed for transmission at 28.8 Kbps. Likewise, if the user clicks on button 114, data sends from a file containing streaming media in a format designed for transmission at 56.6 Kbps. If the user clicks on button 114, the server delivers data from a file containing streaming media in a format designed for transmission at a rate greater than 56.6Kbps and up-to the typical speed of a T1 connection.

However, the primary problem with the "select-a-bandwidth" approach is that it requires a thoughtful selection by a user. This approach invites selection errors.

It requires that a user care, understand, and have knowledge of her connection speed. Often, a user does not pay particular attention to which button to press. The user may only know that a media presentation will appear if the user presses one of these buttons. Therefore, they press any one of them.

Often, a user does not understand the concept of bandwidth. A user may choose button 116 because she may want to see the presentation at its highest quality. This user does not realize that seeing the presentation at its highest quality may result in a non-smooth presentation because her Internet connection cannot handle the rate that the data is being sent through it.

If she does understand the concept of bandwidth, then the user may not know her bandwidth. A user may simply be ignorant of her bandwidth. In addition, varying degrees of noise may cause varying connection speeds each time a user connects to the Internet. Furthermore, some types of connections (such as a cable modem) can have wide degrees of connection speed depending upon numerous factors.

Moreover, the user needs to understand the implications of an incorrect choice. A user needs to be educated so that she understands that she needs to select an option that is equal to or less than her bandwidth to get a smooth presentation. But she should not choose one that is significantly less than her bandwidth. If she does, then she will be seeing a smooth presentation at a lower quality that she could otherwise see at a higher available bandwidth.

As can be seen by the above discussion, this manual approach is often confusing and intimidating to many user. Therefore, it often results in incorrect selections.

What's more, maintaining multiple files (one for each bandwidth) at the media server adds to the overhead of maintaining a Web site.

Automatic Bandwidth Detection

To overcome these problems, media servers use a single file containing subfiles for multiple bandwidths. Also, media servers automatically detect the bandwidth.

This single file is called a MBR (multiple bit rate) file. The MBR files typically include multiple differing "bands" or "streams." These bands may be called "subfiles." A user only clicks on one link. Automatically, behind the scenes, the server determines the right speed band to send to the client.

This automatic speed detection may take a long time. This means that an additional five seconds to a minute (or more) is added to the user's wait for the presentation to begin. This delay for existing automatic speed detection is because of long "handshaking" times while the speed determination is going on.

One existing automatic detection technique involves sending multiple data packets for measuring the speed between the server and client. This technique is described further below in the section titled, "Multiple Measurement Packets Technique."

Bandwidth Measurement Packets

Typically, automatic bandwidth detection techniques measure bandwidth between entities on a network by sending one or more packets of a known size.

Fig. 3 shows a time graph tracking the transmission of two such packets (P_x and P_y) between a sender (e.g., server) and a receiver (e.g., client). The server and client sides are labeled so. On the graph, time advanced downwardly.

Time t_a indicates the time at the server the transmission of P_x begins. Time t_b indicates the time at the server the transmission of P_x ends. Similarly, Time t_0 indicates the time at the client begins receiving P_x . Time t_1 indicates the time at the client completes reception of P_x . At t_1 , the network hardware presumably passes the packet up the communication layers to the application layer.

Packet P_y is similarly labeled on the time graph of Fig. 3. t_c is the server time at the transmission of P_y begins. t_d is the server time that the transmission of P_y ends. Similarly, t_2 the client time that it begins receiving P_y . t_3 is the client time that it completes reception of P_y . At t_3 , the network hardware presumably passes the packet up the communication layers to the application layer.

Bandwidth measurement using a single packet. In a controlled, laboratory-like environment, measuring bandwidth between two entities on a network is

straightforward. To make such a calculation, send a packet of a known size from one entity to the other and measure the transmission latency, which is the amount of time it takes a packet to travel from source to destination. Given this scenario, one must know the time that the packet was sent and the time that the packet arrived.

This technique is nearly completely impractical outside of the laboratory setting. It cannot be used in an asynchronous network (like the Internet) because it requires synchronization between the client and server. Both must be using the same clock.

Alternatively, the client may track the time it begins receiving a packet (such as t_0 for P_x) and the time the packet is completely received (such as t_1 for P_x).

Fig. 3 shows packet P_x being sent from a server to a client. P_x has a known size in bits of PS. The formula for calculating bandwidth (bw) is

$$bw(P_x) = \frac{PS}{t_1 - t_0}$$

Formula 1 (Single Packet)

This technique works in theory, but unfortunately does not work in practice. Only the hardware knows when a packet is initially received. Therefore, only the hardware knows when t_0 is.

The other communication layers (such as the transport layer and the application layer) can only discover the time when the packet is completely received by the hardware. That is when the hardware passes it up to them. This

1 completion time for packet P_x is t_1 . It is not possible to calculate bandwidth only
2 one knowing one point in time.

3
4 Packet-pair. A technique called packet-pair is used to overcome these
5 problems in asynchronous networks. With packet-pair, two identical packets are
6 sent back-to-back. The server sends a pair of packets, one immediately after the
7 other. Both packets are identical; thus, they have the same size (PS). The
8 bandwidth is determined by dividing the packet size by the time difference in
9 reception of each packet.

10 Each packet has specific measurable characteristics. In particular, these
11 characteristics include its packet size (PS) and the measured time such a packet
12 arrives (e.g., t_{0-3} in Fig. 3). Some characteristics (such as packet size) may be
13 specified rather than measured, but they may be measured if so desired.

14 As shown in Fig. 3, the server sends packet, P_x . The client's hardware
15 begins receiving the packet at t_0 . When reception of the packet is complete at t_1 ,
16 the hardware passes it up the communication layers. Ultimately, it is received by
17 the destination layer (e.g., application layer) at presumably t_1 .

18 After the server sends P_x (which completed at t_b), it immediately sends
19 packet P_y at t_c . It is important that there be either 1) absolutely no measurable
20 delay between t_b and t_c or 2) a delay of a known length between t_b and t_c . Herein,
21 to simplify the description, it will be assumed that there is no measurable delay
22 between t_b and t_c .

23 The client's hardware begins receiving P_y at t_2 . When reception of the
24 packet is complete at t_3 , the hardware passes it up the communication layers.
25

1 Ultimately, it is received by the destination layer (e.g., application layer) at
2 presumably t_3 .

3 Fig. 3 shows no delay between t_1 (the time of completion of reception of P_x)
4 and t_2 (the time reception of P_y begins). Theoretically, this will always be the case
5 if P_x and P_y are transmitted under identical conditions. In practice, is the often the
6 case because P_y is sent immediately after P_x .

7 Using packet-pair, the formula for calculating bandwidth (bw) is

$$\text{bw}(P_x P_y) = \frac{PS}{t_3 - t_1}$$

8
9
10
11 **Formula 2 (Packet-Pair)**

12 This technique works in theory and in practice. However, it only works
13 well over a network that is relatively static.

14 For example, in Fig. 1, assume the network consists of only the server 20;
15 routers 32, 34, and 36; a specific ISP of ISPs 80; and client 90. Further, assume
16 that the links between each node on this static network is fixed and has a
17 consistent bandwidth. In this situation, the packet-pair techniques provide an
18 accurate and effective measurement of bandwidth.

19
20 Packet-pair does not work well over the Internet. However, the packet-pair
21 technique does not work well over a dynamic network, like the Internet. A
22 dynamic network is one where there is a possibility that a packet may be handled
23 in a manner different from an earlier packet or different from a later packet.
24
25

1 Fig. 1 illustrates examples of those handling differences. Assume that all
2 packets are traveling from the server to the client (from left to right in Fig. 1).
3 Assume that packets 60-68 were sent back-to-back by the server 20 to the client
4 90. Assume that packet 70 was sent by another server (not shown) to the client 90
5 and it is unrelated to bandwidth measurement.

6 Notice, as illustrated in Fig. 1, that packets may take different routes. In
7 addition, some routes may significantly delay the packet transmission. This is
8 especially true if the packet is transmitted via an apparently unusual (but not
9 necessarily uncommon) route, such as wireless transmission, overseas via an
10 underwater cable, satellite transmission (as shown by dishes 46 and 50 and
11 satellite 48), etc.

12 A router (such as router 42) may delay a packet (such as 64) more than
13 another may by temporarily buffering it. Another packet (such as packet 70) from
14 another source may slip in between two packets (such as packets 60 and 62). In
15 addition, a modem (not shown) of the client may compress packets.

16 Communications equipment (such as a modem) may compress a packet
17 (such as 66) to shrink the packet size and thus speed along transmission. Such
18 packet compression can significantly affect the bandwidth measurement because
19 not all of the subsequent data packets will be compressed or compressed at the
20 same rate.

21 22 **Multiple Measurement Packets Technique**

23 To overcome these problems, conventional automatic bandwidth
24 measurement techniques uses multiple packets. A server sends several (much
25

more than two) packets and calculates the speed of each. Conventional wisdom on bandwidth measurement indicates that in order to get accurate measurements several pairs of packets must be sent repeatedly over several seconds to several minutes. Herein, this technique is called "multiple-packets" to distinguish it from the above-described "packet-pair" technique.

Typically, the ultimate bandwidth is determined by finding the average of the many bandwidth measurements. This averaging smoothes out variances in delays for each packet; however, it does not compensate for packet compression during transmission. One of two extremely incorrect measurements will skew the average.

Unfortunately, this technique takes a long time relative the existing wait for the user between click and media presentation. A long time may be five seconds to several minutes depending on the data and the situation. Such a delay adds to the annoyance factor for the user who wishes experience the media presentation. This is not an acceptable delay. Since there are no other options available using conventional techniques, the user has be forced to endure these delays.

Moreover, these conventional approaches typically use TCP to transmit the packets. Using TCP introduces additional delays for handshaking. These conventional approaches typically modify the kernel of the operating system (usually the transport layer) to do perform these measurements.

No existing automatic bandwidth measurement can nearly instantaneously measure bandwidth across the Internet using a pair of packets. No existing automatic bandwidth measurement can make such measurements at the application layer. Thus, it avoids modifying the operating system. No existing

1 automatic bandwidth measurement addresses measurement distortion caused by
2 packet compression.

3 4 5 SUMMARY

6 The fast dynamic measurement of connection bandwidth utilizes a single
7 pair of packets to calculate bandwidth between two entities on a network (such as
8 the Internet). This calculation is based upon the packet-pair technique. This
9 bandwidth measurement is extremely quick.

10 On its journey across a network, communication equipment and modems
11 may compress a packet. This compression shrinks the size of the packet; thus, it
12 can distort the bandwidth calculation using such a shrunken packet. To avoid this
13 distortion, the fast dynamic measurement of connection bandwidth employs non-
14 compressible packets. More specifically, it employs highly entropic packets.
15 Therefore, a packet cannot be compressed during its journey.

16 In addition, on its journey across a network, packets may be rerouted,
17 delayed, misrouted, and the like. These momentary delays may result in a
18 momentary bad bandwidth calculation. This problem is ameliorated by using a
19 history list at the client that keeps track of recent measurements. The client
20 returns the median of that list to the server. That median is the specified
21 bandwidth.

1 **BRIEF DESCRIPTION OF THE DRAWINGS**

2 Fig. 1 illustrates a typical public networking environment (such as the
3 Internet) and the routing of and delay of data packets sent from a server to a client.

4 Fig. 2 is cut-away portion of a Web page. The cut-away shows a user
5 interface providing a user a mechanism for selecting the bandwidth. This shows a
6 conventional technique for determining bandwidth.

7 Fig. 3 shows a packet pair (being sent from a server to a client) graphed in
8 the time domain. This shows a conventional implementation of packet-pair
9 technique to measure bandwidth.

10 Fig. 4 also illustrates a typical public networking environment (such as the
11 Internet). This shows a pair of packets sent back to back.

12 Fig. 5 is visual representation of a history list of the last ten measured
13 bandwidths.

14 Fig. 6 is a flowchart illustrating the methodology of an implementation of
15 the server side of the exemplary bandwidth meter.

16 Fig. 7 is a flowchart illustrating the methodology of an implementation of
17 the client side of the exemplary bandwidth meter.

18 Fig. 8 is an example of a computing operating environment capable of
19 implementing the server side and/or the of the exemplary bandwidth meter.

20
21 **DETAILED DESCRIPTION**

22 The following description sets forth a specific embodiment of the fast
23 dynamic measurement of connection bandwidth that incorporates elements recited
24 in the appended claims. This embodiment is described with specificity in order to
25

1 meet statutory written description, enablement, and best-mode requirements.
2 However, the description itself is not intended to limit the scope of this patent.
3 Rather, the inventors have contemplated that the claimed fast dynamic
4 measurement of connection bandwidth might also be embodied in other ways, in
5 conjunction with other present or future technologies.

6 Even when used with a public network (such as the Internet), an exemplary
7 fast dynamic measurement of connection bandwidth (i.e., "bandwidth meter" or
8 "bw-meter") described herein is fast and robust. The exemplary bandwidth meter
9 implements a low-latency technique for automatically measuring the network
10 bandwidth available between two entities on a communications network. It has
11 been found to be particularly useful over the Internet (or other such public
12 networks).

13 Unlike the conventional approaches, the exemplary bw-meter obtains a best
14 effort bandwidth measurement with the least possible delay, even under difficult
15 network conditions. The exemplary bw-meter is designed to provide reasonable
16 output in less than one second in most existing networks, including LANs, cable,
17 DSL, and modem connections.

18 Furthermore, the exemplary bw-meter is implemented at the application
19 layer (rather than other layers like the transport and physical layers). Although the
20 exemplary bw-meter may be implemented on other layers, the one described
21 herein is implemented on the application layer. In particular, it may be partially
22 implemented by a Web browser or a media player.
23
24
25

Packet-Pair Technique

The exemplary bw-meter utilizes the well-established packet-pair technique described above and illustrated in Fig. 3. The exemplary bw-meter uses the packet-pair formula (Formula 2) described above to calculate the maximum de facto bandwidth between two entities on a communications network (such as the Internet).

Unlike existing automatic bandwidth measurement techniques that use multiple packets, the exemplary bw-meter uses a single pair of packets for measuring bandwidth over the Internet. With the exemplary bw-meter, bandwidth measurements and calculations are made “nearly instantaneously” because only a single pair of measurement packets is sent. The term “nearly instantaneously” means that the bandwidth is determined as soon as the pair of packets arrive at the client.

The exemplary bw-meter overcomes the drawbacks and limitations of using packet-pairs over the Internet by employing non-compressible packets and a history list.

Non-Compressible Packets

Data compression. Storing data in a format that requires less space than it would otherwise is data compression. Alternatively, it is transmitting data in a format that requires less bandwidth than it would otherwise.

Data compression is particularly useful in communications because it enables devices to transmit the same amount of data in fewer bits. There is a

1 variety of data compression techniques. Some are lossless (no data loss by
2 compression) and some are lossy (some data loss by compression).

3 In general, lossless techniques are used in communication. These
4 techniques compress data by representing long data patterns by short codes. Thus,
5 some data is more compressible than others are.

6
7 Entropy. A measure of the inherent compressible characteristics of a data is
8 “entropy.” It may be defined to be a measure of the number of bits necessary to
9 transmit (or store) data as a function of the probability that the message will
10 consist of a specific set of symbols (i.e., recognizable data patterns).

11 Data with low entropy means that the data has many recognizable patterns
12 and it may be significantly compressed. Data with high entropy means that the
13 data has few (to zero) recognizable patterns and it may be minimally (to zero)
14 compressed. Herein, “highly entropic” data is data that contains nearly no
15 recognizable patterns; thus, compression of such data is nearly identical in size to
16 the data when uncompressed.

17
18 Contents of packets contain no information. For measuring bandwidth, the
19 contents of the packets contain no information. If the packets used for measuring
20 bandwidth remain the same size throughout transmission, the actual contents of
21 the packets do not affect the measurement.

22 Since the content carries no semantic information (meaning), the packet’s
23 contents may be any data, regardless of meaning, that is not compressible. More
24
25

specifically, the data of the packets of the exemplary bw-meter are highly entropic to prevent (or greatly reduce the affect of) data compression during transmission.

Alternatively, the content of the packets may contain information useful to the client or server, but that information will be highly entropic.

Packet Containing Highly Entropic Data. With the exemplary bw-meter, the packets sent include highly entropic data. This means that the data in the packets are not compressible.

One of the problems of using a conventional packet-pair technique is that some modems and some communications equipment compress the packets. If this happens, the bandwidth calculation will be off because the size of the packets may have been alternated during the transmission. If the packets are highly entropic, then the packets cannot be compressed; thus, the results of the bandwidth calculations can be relied upon.

A packet containing highly entropic data (i.e., "highly entropic packet") of the packet-pair technique of the exemplary bw-meter travels from the sending entity (e.g., server) to the receiving entity (e.g., client). Since it is highly entropic, it is not successfully compressed along its journey. Fig. 4 shows an example of such a journey. Fig. 4 illustrates an environment similar to that shown in Fig. 1.

Fig. 4 illustrates an example of a typical Internet configuration. It includes a server (such as media server 220), which is coupled to the Internet 230. The server typically includes one or more physical server computers 222 with one or more physical storage devices and/or databases 224. On the other side of an

Internet transmission is a client 290, which is connected via its Internet Service Providers (ISPs) 280.

Cloud 230 is labeled the Internet, but it is understood that this cloud represents that portion of the Internet that does not include the server, client's ISPs, and the client. Inside such cloud are the routers, transmission lines, connections, and other devices that more-often-than-not successfully transmit data between clients and servers. Inside exemplary Internet cloud 230 are routers 232-244; two satellite dishes 246 and 250; and a satellite 248. These represent the possible paths that a data packet may take on its way between the server and the client.

Fig. 4 shows successive packets 260 and 262 of the pair sent in accordance with the exemplary bw-meter. The server 220 sends packet 262 immediately after packet 260. As shown in Fig. 4, neither packet 260 nor 262 are compressed.

Adaptable Compression Countermeasure. In addition, the entropic data included in the packets is regularly changed. Some communications equipment and modems may adaptively compress otherwise uncompressible data if it sees the exact same data repeatedly. These devices "learn" how to compress entropic data. They do this by storing the data and using a short code to represent it.

Therefore, the entropic content of the packets is altered regularly so that adaptable compression techniques cannot "compress" the otherwise non-compressible data. In the exemplary bw-meter, a set of 20-50 different entropic packets is regularly used.

Large Packets

A packet's size is based upon the number of data bits contained therein. It is preferable to use large packets to measure bandwidth. It is analogous to getting better statistical data when one has a larger sample. However, if a packet becomes too large, it will be fragmented during transmission by communications equipment. If any fragment of a packet is delayed, then the timing (e.g., t_0 - t_3 as shown in Fig. 3) may be off.

Thus, it is preferable to avoid fragmentation. Therefore, the exemplary bw-meter employs a package with a size as large enough as possible but just small enough to avoid fragmentation. Herein, this size is called "fragmentation-avoidance" size or simply "frag-avoid." Thus, this frag-avoid size refers to the largest size possible before a packet is divided during transmission.

For example, if it is known that any packet over 1000 bits is fragmented but none below 1000 are fragment, then the fragmentation-avoidance size is 999 bits.

UDP packets. With the exemplary bw-meter, the packet pair is composed of a 1472 byte UDP packet immediately followed by a 1472 byte UDP (User Datagram Protocol) packet. Both packets contain highly entropic data to guarantee that the packets remain 1472 bytes long across all the hops in the network.

Although UDP is preferred, the packets (such as measurement packets 260 and 262 in Fig. 4) may be any protocol format. When sent via the Internet, the measurement packets of the exemplary bw-meter are sent via TCP or UDP.

TCP (Transmission Control Protocol) is one of the main protocols in TCP/IP networks (such as the Internet). Whereas the IP protocol deals only with

005000-4000000

1 packets, TCP enables two hosts to establish a connection and exchange streams of
2 data. TCP guarantees delivery of data and guarantees that packets will be
3 delivered in the same order in which they were sent.

4 UDP is a connectionless protocol that (like TCP) runs on top of IP
5 networks. Unlike TCP/IP, UDP/IP provides very few error recovery services,
6 offering instead a direct way to send and receive packets (i.e., datagram) over an
7 IP network.

8 With the exemplary bw-meter, UDP packets are preferred over TCP
9 packets. Some applications handle TCP packets in a non-standard fashion. An
10 example of such an application is the software from America Online (AOL).
11 Instead of delivering the packets to the application upon receipt, the AOL software
12 delivers packets at specific time intervals. This time-delivery granularity greatly
13 distorts the bandwidth measurement. These non-standard applications, like AOL,
14 do not handle UDP packets in such a manner.

15 Furthermore, TCP includes additional administrative traffic that UDP does
16 not include. In TCP, when a packet is sent, the client must send an
17 acknowledgement (i.e., "ack"). The kernel (i.e., the transport layer) of the
18 operating system handles this acknowledgement process.

19 For example, suppose that the second packet is lost. Since the client did not
20 receive a second packet, it does not send an ack. Since the server receives no ack,
21 the server sends the second packet again. Therefore, the application eventually
22 receives its expected second packet. However, it has no way to know that this is a
23 resend of the second packet. The bandwidth measurement will be distorted
24 because of the delay introduced by a packet loss and resend. With TCP packets, it
25

is possible to get a bad measurement and have no way of knowing that it is a bad measurement. That is a bad combination.

With UDP, when a packet is lost, the client never gets it. Thus, the application knows if the measurement is bad because it never gets the second packet. UDP can be controlled at the application level better than TCP can.

When exemplary bw-meter is used across a firewall, TCP packets are preferred over UDP. Most firewalls do not allow UDP packets to flow across them.

History List

The nearly instantaneous bandwidth measurement described above is quick and efficient. As shown in Fig. 1, packets may be delayed for many reasons. These cause of these delays and the amount of the delays may vary from moment to moment. Thus, this bandwidth measurement at a particular moment in time may not reflect the actual bandwidth typically experienced. The calculated bandwidth for each measurement may be slightly off for various reasons. Occasionally, it will be extremely off.

Fig. 5 shows an example of a history list 300 of recorded bandwidth measurement. The client keeps this history of calculated bandwidths to ameliorate the affect of such momentary delays. The exemplary bw-meter uses the median of the history of bandwidth as the proper speed. Using the history list lessens the impact of one or two erroneous measurement.

Items are added and removed from the list in a LIFO (last in, first out) fashion. Therefore, the last ten measurements are always included in the list.

1 Experience has shown that the median of a ten-item history tends to give a
2 good approximation of the actual bandwidth. Although the list can be any size
3 that gives a good approximation of the actual bandwidth.

4 The history list 300 of Fig. 5 includes ten items. Each item is a recorded
5 bandwidth measurement taken when the client accessed a media file. The median
6 of this list is 46Kbps as shown at item 302 and 304. Therefore, a server will
7 choose a media file that is formatted to be sent at 46Kbps or slower.

8 Instead of using the median, the exemplary bw-meter may use some other
9 statistical derivation, such as an average. The median is preferred over the
10 average because the median is less likely to be skewed by an extremely incorrect
11 (too low or too high) entry.

12 With a history list, a good estimation of the link bandwidth may be
13 calculated very quickly, after receiving just one packet pair. Therefore, a
14 presentation may be presented right away. Conventional multi-packet approaches
15 need to send many packet pairs one after the other. This results in a significant
16 delay before for the player starts displaying a video presentation once the user
17 clicks "play." This is usually very annoying for users, so it is not acceptable.

18
19 Flushing the History List. The history list is completely cleared (i.e.,
20 "flushed") from time to time. In particular, this is done each time a new
21 connection to the Internet is established. This can be determined by the use of a
22 new IP address.

23 Each time a modem connects to the Internet the speed can be significantly
24 different from the recent past. Many factors, such as line noise, affect this
25

005099"4099990
09636304"096990
1 connection speed. For example, assume that a client has a history list like that
2 shown in Fig. 5. On the next connection, user is given a new IP address. Also, the
3 modem connects at 26Kbps and replaced item 302 with 26. The media is still
4 46Kbps, but the current speed is actually 26Kbps. Therefore, the server will
5 overrun the modem when it sends a file at 46Kbps.

6 Therefore, flushing the history list upon an IP address change alleviates this
7 problem.

8 This problem typically does not occur in LANs (Local Area Networks),
9 DSL, cable modem, or other similar dedicated Internet connections. In these
10 situations, the IP addresses rarely if ever change.

11 12 **Modem Query**

13 If the exemplary bw-meter gets a measurement that is below a low-
14 believability threshold or above a high-believability threshold, it will disregard the
15 results. These thresholds define the "range of believability" for the results. If the
16 result is below the low-believability threshold, the exemplary bw-meter will not
17 believe that the bandwidth is that low. Likewise, if the result is above the high-
18 believability threshold, the exemplary bw-meter will not believe that the
19 bandwidth is that high.

20 In the exemplary bw-meter, low-believability threshold is 24.4 Kbps and
21 the high-believability threshold is 1 Mbps. The range of possible measurements
22 between these thresholds is called the "believability" range.
23
24
25

1 The following exemplary pseudocode restricts a variable called
2 "HistoryMedian" to possible values that it can take. Typically, the range is from
3 24,000 to 1,000,000.

```
4  
5       if (HistoryMedian > 1,000,000)  
6       {  
7       return UnlimitedBandwidth;   //needed for OS jitter issues  
8       }  
9       else  
10      {  
11      return max (24,000, HistoryMedian);   //ensure a floor of  
12      24,000 is always set  
13      }
```

14 There is a registry entry (HardCodedRegistryValue) that, if present,
15 takes precedence over the output of the exemplary bw-meter .
16

17 When the result fall outside the believability range, the exemplary bw-
18 meter asks the modem of the client for its speed. Also, in this situation, the
19 median of the history list will still be used as the proper bandwidth measurement,
20 but the query of the modem will provide the new entry into the list.
21

22 The exemplary bw-meter may ask the operating system (OS) of the client.
23 The OS tends to give poor bandwidth estimations. Thus, the exemplary bw-meter
24 asks the modem (using TAPI). The modems do not give great bandwidth
25 estimations, but they tend to be better than the OS.

It has been discovered that some modems react poorly when asked for their
bandwidth measurement. Examples of poor reactions include system crashes.

To avoid this with certain operating systems (such as Windows 95®,
Windows 98®, Windows NT®, Windows 2000® by the Microsoft Corporation),

1 the exemplary bw-meter specifically looks for an adapter with a MAC address
2 starting with "44-45 ". This indicates the presence of a PPP device. If this exists,
3 the exemplary bw-meter calls TAPI. Otherwise, the exemplary bw-meter does not
4 call TAPI.

5 6 **Methodological Implementation**

7 Server Side. Fig. 6 shoes a methodological implementation of the server
8 side of the exemplary bandwidth meter. At 300, the dynamic bandwidth
9 measurement in accordance with the exemplary bandwidth meter is initiated.
10 Typically, a user of the client selects an option on a Web page to experience a
11 media presentation. Alternatively, an application on the client may initiate such
12 bandwidth measurement. Such an application may be a Web browser, media
13 player, or the like.

14 At 302 of Fig. 6, the server selects a highly entropic packet from at set of
15 such packets. The server has a set of packets from which to choose. Each packet
16 in the set is highly entropic. This selection may be via any fashion, such as
17 revolving and random. The packet may be selected at that moment or it may have
18 been pre-selected. Alternatively, the highly entropic packet may be calculated
19 using formula known to generate entropic data.

20 At 304, the server sends a pair of packets to the client, with one
21 immediately following the other. These packets are identical and highly entropic.
22 Alternatively, the packets may be different, but they will have the same size. Each
23 packet is preferably frag-avoid size.
24
25

1 At 306, the server waits for a response from the client. If it is not received
2 within time limit, the process returns to sending a pair of packets at 304.
3 Alternatively, the server may send a pair of packets without waiting for a response.
4 Although not shown in the flowchart, the process will repeat this a given number
5 of times before terminating and generating an error. If a response is received
6 within the time limit, the process proceeds to the next block at 308.

7 The response includes a bandwidth measurement determined by the client
8 using the pair of packets sent by the server at 304. The server extracts the specified
9 bandwidth from the response at 308.

10 At 310 of Fig. 6, the server selects the file (or portion thereof) formatted for
11 a bandwidth equal to or just lesser than the specified bandwidth. At 312, the server
12 sends the file (or portion thereof) to the client.

13 If it was a media, file the user of the client enjoys a media presentation that
14 begins play quickly. It also plays smoothly and at the highest quality possible at a
15 measured bandwidth. The process ends at 314.

16 Client Side. Fig. 7 shoes a methodological implementation of the client
17 side of the exemplary bandwidth meter. At 350, the dynamic bandwidth
18 measurement in accordance with the exemplary bandwidth meter is initiated. This
19 initiation occurs in the same as described above for initiation from the server
20 perspective.

21 At 352, the client receives a first packet of packet-pair. This is like packet
22 P_x in Fig. 3. The client notes and stores the time that it receives this packet. This
23 is like time t_1 shown in Fig. 3.
24
25

At 354 of Fig. 7, the client receives a second packet of packet-pair. This is like packet P_y in Fig. 3. The client notes and stores the time that it receives this packet. This is like time t_3 shown in Fig. 3.

At 356 of Fig. 7, the client determines the packet size (PS) of the packets. This may be known constant. This may be provided by handshaking data between the server and the client. This may be encoded in the contents of the packets. The client may simply measure this.

At 358, the client calculates the measured bandwidth using the packet-pair formula (Formula 2) above. Such formula is repeated here:

$$bw(P_x P_y) = \frac{PS}{t_3 - t_1}$$

Formula 2 (Packet-Pair)

At 360, the client determines if the calculated bandwidth is inside range of believability. This is the range of measurements that are believable. It is defined to be between the low-believability threshold (e.g., 24.4 Kbps) and the high-believability threshold (e.g., 1 Mbps).

If the client determines that the calculated bandwidth is inside range of believability, then the process proceeds to block 380 described below.

If it is outside the range, then the client queries the operating system (OS) and/or modem to determine what they believe the current bandwidth to be. If current bandwidth specified by the OS and/or the modem is outside the range of believability, then set bandwidth to be a believable value. For example, set the bandwidth to the low-believability threshold (e.g., 24.4 Kbps) if it is below that

threshold. Alternatively, set the bandwidth to the high-believability threshold (e.g., 1 Mbps) if the it is above that threshold. Go to block 380 described below

If current bandwidth specified by the OS and/or the modem is inside the range of believability, then go to block 380 described below.

At 380 in Fig. 7, the client stores the bandwidth into the history list. The new entry replaces the entry that has been in the list the longest. If this is a new connection, the client flushes the list and enters the bandwidth as the only entry.

At 382 and 384, the client finds the median of history list and returns it to the server. At 386, the process ends.

Additional Exemplary Implementation Details

On the server side, the exemplary bw-meter is designed to run in one of three modes: MMSU (Microsoft Multimedia Server UDP), MMST (Microsoft Multimedia Server TCP), and HTTP (Hypertext Transport Protocol). The MMSU mode is used when the connection between the server and the client is such that the media packets can be sent over UDP. In other words, when there is no proxy computer between the server and the player, or, if there is one, when the proxy computer lets UDP traffic go across. The MMST mode is used if the firewall allows TCP traffic to port 1755, but does not allow any kind of UDP traffic. Finally, the HTTP mode is used if the firewall only allows TCP traffic on port 80.

1 The following is examples of pseudocode that a client may use when
2 communicating with a server in MMSU mode:

3
4 if (HardCodedRegistryValue)
5 {
6 return HardCodedRegistryValue;
7 }
8 else If (PP measurement successful)
9 {
10 If PP < 24Kbps, save 24Kbps in history, and then
11 return HistoryMedian;
12 else save PP measurement in history, and then return
13 HistoryMedian;
14 }
15 else //Case where UDP PP fails
16 {
17 else if history is not empty, return HistoryMedian;
18 else if TAPI supported device, return (TAPI - 14%);
19 else return UnlimitedBandwidth;
20 }
21
22
23
24
25

1 The following is examples of pseudocode that a client may use when
2 communicating with a server in MMST mode:

3
4 if (HardCodedRegistryValue)
5 {
6 return HardCodedRegistryValue;
7 }
8 else if HistoryMedian exists, return HistoryMedian;
9 else if TAPI supported device
10 {
11 if (V4.1 server)
12 {
13 TCP based PP measurement and save to history;
14 *//since packets are high entropy, may be good*
15 *measurement*
16 return max (24,000, HistoryMedian);
17 }
18 else *//v4.0 and v3 server*
19 {
20 if HistoryMedian exists, return max (24,000,
21 HistoryMedian); *//We don't trust PP sample*
22 else return TAPI - 14%
23 }
24 }
25 else if (V4.1 Server || V4.0 Server)
26 {
27 TCP based PP measurement and save to history; *//not a*
28 *modem, entropy not a concern*
29 return max (24,000, HistoryMedian));
30 }
31 else if HistoryMedian exists, return HistoryMedian;
32 else return UnlimitedBandwidth;

1 The following is examples of pseudocode that a client may use when
2 communicating with a server in HTTP mode:

```
3  
4     if (HardCodedRegistryValue)  
5     {  
6         return HardCodedRegistryValue;  
7     }  
8     else if history is not empty, return HistoryMedian;  
9     else if TAPI supported device, return (TAPI - 14%);  
10    else return unlimited bandwidth;
```

11 **Exemplary Computing Environment**

12 Fig. 8 illustrates an example of a suitable computing environment 920 on
13 which the exemplary bw-meter may be implemented.

14 Exemplary computing environment 920 is only one example of a suitable
15 computing environment and is not intended to suggest any limitation as to the
16 scope of use or functionality of the exemplary bw-meter. Neither should the
17 computing environment 920 be interpreted as having any dependency or
18 requirement relating to any one or combination of components illustrated in the
19 exemplary computing environment 920.

20 The exemplary bw-meter is operational with numerous other general
21 purpose or special purpose computing system environments or configurations.
22 Examples of well known computing systems, environments, and/or configurations
23 that may be suitable for use with the exemplary bw-meter include, but are not
24 limited to, personal computers, server computers, thin clients, thick clients, hand-
25 held or laptop devices, multiprocessor systems, microprocessor-based systems, set
top boxes, programmable consumer electronics, wireless phones, wireless

communication devices, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The exemplary bw-meter may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The exemplary bw-meter may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

As shown in Fig. 8, the computing environment 920 includes a general-purpose computing device in the form of a computer 930. The components of computer 920 may include, by are not limited to, one or more processors or processing units 932, a system memory 934, and a bus 936 that couples various system components including the system memory 934 to the processor 932.

Bus 936 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA)

006030-40000000

1 local bus, and Peripheral Component Interconnects (PCI) buss also known as
2 Mezzanine bus.

3 Computer 930 typically includes a variety of computer readable media.
4 Such media may be any available media that is accessible by computer 930, and it
5 includes both volatile and non-volatile media, removable and non-removable
6 media.

7 In Fig. 8, the system memory includes computer readable media in the form
8 of volatile, such as random access memory (RAM) 940, and/or non-volatile
9 memory, such as read only memory (ROM) 938. A basic input/output system
10 (BIOS) 942, containing the basic routines that help to transfer information
11 between elements within computer 930, such as during start-up, is stored in ROM
12 938. RAM 940 typically contains data and/or program modules that are
13 immediately accessible to and/or presently be operated on by processor 932.

14 Computer 930 may further include other removable/non-removable,
15 volatile/non-volatile computer storage media. By way of example only, Fig. 8
16 illustrates a hard disk drive 944 for reading from and writing to a non-removable,
17 non-volatile magnetic media (not shown and typically called a "hard drive"), a
18 magnetic disk drive 946 for reading from and writing to a removable, non-volatile
19 magnetic disk 948 (e.g., a "floppy disk"), and an optical disk drive 950 for reading
20 from or writing to a removable, non-volatile optical disk 952 such as a CD-ROM,
21 DVD-ROM or other optical media. The hard disk drive 944, magnetic disk drive
22 946, and optical disk drive 950 are each connected to bus 936 by one or more
23 interfaces 954.

1 The drives and their associated computer-readable media provide
2 nonvolatile storage of computer readable instructions, data structures, program
3 modules, and other data for computer 930. Although the exemplary environment
4 described herein employs a hard disk, a removable magnetic disk 948 and a
5 removable optical disk 952, it should be appreciated by those skilled in the art that
6 other types of computer readable media which can store data that is accessible by a
7 computer, such as magnetic cassettes, flash memory cards, digital video disks,
8 random access memories (RAMs), read only memories (ROM), and the like, may
9 also be used in the exemplary operating environment.

10 A number of program modules may be stored on the hard disk, magnetic
11 disk 948, optical disk 952, ROM 938, or RAM 940, including, by way of example,
12 and not limitation, an operating system 958, one or more application programs
13 960, other program modules 962, and program data 964.

14 A user may enter commands and information into computer 930 through
15 input devices such as keyboard 966 and pointing device 968 (such as a "mouse").
16 Other input devices (not shown) may include a microphone, joystick, game pad,
17 satellite dish, serial port, scanner, or the like. These and other input devices are
18 connected to the processing unit 932 through an user input interface 970 that is
19 coupled to bus 936, but may be connected by other interface and bus structures,
20 such as a parallel port, game port, or a universal serial bus (USB).

21 A monitor 972 or other type of display device is also connected to bus 936
22 via an interface, such as a video adapter 974. In addition to the monitor, personal
23 computers typically include other peripheral output devices (not shown), such as
24
25

1 speakers and printers, which may be connected through output peripheral interface
2 975.

3 Computer 930 may operate in a networked environment using logical
4 connections to one or more remote computers, such as a remote computer 982.
5 Remote computer 982 may include many or all of the elements and features
6 described herein relative to computer 930.

7 Logical connections shown in Fig. 8 are a local area network (LAN) 977
8 and a general wide area network (WAN) 979. Such networking environments are
9 commonplace in offices, enterprise-wide computer networks, intranets, and the
10 Internet.

11 When used in a LAN networking environment, the computer 930 is
12 connected to LAN 977 network interface or adapter 986. When used in a WAN
13 networking environment, the computer typically includes a modem 978 or other
14 means for establishing communications over the WAN 979. The modem 978,
15 which may be internal or external, may be connected to the system bus 936 via the
16 user input interface 970, or other appropriate mechanism.

17 Depicted in Fig. 8, is a specific implementation of a WAN via the Internet.
18 Over the Internet, computer 930 typically includes a modem 978 or other means
19 for establishing communications over the Internet 980. Modem 978, which may
20 be internal or external, is connected to bus 936 via interface 970.

21 In a networked environment, program modules depicted relative to the
22 personal computer 930, or portions thereof, may be stored in a remote memory
23 storage device. By way of example, and not limitation, Fig. 8 illustrates remote
24 application programs 989 as residing on a memory device of remote computer
25

1 982. It will be appreciated that the network connections shown and described are
2 exemplary and other means of establishing a communications link between the
3 computers may be used.
4

5 **Exemplary Operating Environment**

6 Fig. 8 illustrates an example of a suitable operating environment 920 in
7 which the exemplary bw-meter may be implemented. Specifically, the exemplary
8 bw-meter is implemented by any program 960-962 or operating system 958 in Fig.
9 8.

10 The operating environment is only an example of a suitable operating
11 environment and is not intended to suggest any limitation as to the scope of use of
12 functionality of the bw-meter described herein. Other well known computing
13 systems, environments, and/or configurations that may be suitable for use with the
14 bw-meter include, but are not limited to, personal computers, server computers,
15 hand-held or laptop devices, multiprocessor systems, microprocessor-based
16 systems, programmable consumer electronics, network PCs, minicomputers,
17 mainframe computers, distributed computing environments that include any of the
18 above systems or devices, and the like.
19

20 **Computer-Executable Instructions**

21 An implementation of the exemplary bw-meter may be described in the
22 general context of computer-executable instructions, such as program modules,
23 executed by one or more computers or other devices. Generally, program modules
24 include routines, programs, objects, components, data structures, etc. that perform
25

particular tasks or implement particular abstract data types. Typically, the functionality of the program modules may be combined or distributed as desired in various embodiments.

Computer Readable Media

An implementation of the exemplary bw-meter may be stored on or transmitted across some form of computer readable media. Computer readable media can be any available media that can be accessed by a computer. By way of example, and not limitation, computer readable media may comprise computer storage media and communications media.

Computer storage media include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by a computer.

Communication media typically embodies computer readable instructions, data structures, program modules, or other data in a modulated data signal such as carrier wave or other transport mechanism and included any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the

signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media. Combinations of any of the above are also included within the scope of computer readable media.

Conclusion

The exemplary bw-meter is superior to conventional approaches because it is faster and more accurate. It is faster to send one pair of packets to measure bandwidth than multiple packets. It is more accurate because the packets are not compressible and it includes a history list. This exemplary bw-meter has been extensively tested in many different types of networks. In more than ninety percent (90%) of the cases, the results are within ten percent (10%) of the expected bitrates.

Although the fast dynamic measurement of connection bandwidth has been described in language specific to structural features and/or methodological steps, it is to be understood that the fast dynamic measurement of connection bandwidth defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed fast dynamic measurement of connection bandwidth.